Release 1.0

86-3 14 March 1986

RISC TECHNOLOGY

The most exciting buzzword of the season is RISC technology: It is spice both for IBM's low-end RT PC and HP's new high-end Spectrum line, compatible with its existing 3000 series and other HP machines.

Very simply, RISC technology lets microprocessors run faster, but its *significance* depends on specific design decisions -- which were at opposite ends of the spectrum (sorry!) for HP and for IBM. The RT PC may have gotten its funding, and will achieve fame, for its RISC technology, but it will sell on the basis of its dual-bus architecture, its breadth of software, and its powerful memory management, which makes it ideally suited for symbolic processing. HP's Spectrum will sell to the extent that it restores price-performance to the large and well-loved suite of HP 3000 applications.

Still, let's briefly consider RISC technology. We like to style it no-frills technology: You get only what you pay for. Currently, most standard microprocessors have far more instructions than most people ever use. As logic kept getting cheaper, designers added functions because they looked so cost-effective -- at least in terms of previously prevailing prices. The resulting microprocessors were highly optimized for a huge number of tasks they rarely performed. Naturally enough, this specificity had its costs: Users were paying in dollars and performance for functions they never used. Chips were loaded down with complex instructions that hampered processing and wasted costly microprocessor real estate.

While this was going on in market-driven areas, where each new frill was a potential selling point, IBM and HP researchers were studying the instructions that most folks actually used on their computers. Naturally enough, they found that only 20 percent of the instructions accounted for

⟶

HAPPY BIRTHDAY, GEORGE!

INSIDE

RISC TECHNOLOGY 1
Two flavors.
FACTS AS ARTIFACTS 7
Orders of knowledge.
THE OMNIVOROUS MCChine 10
The soul of CYC.
Fred's Pumps & Parts.
ARTIFICIAL WEALTH 18
Teknowledge goes public.
Oracle on a roll.
CD-ROMFEST 20
Legitimizing the market.
RELEASE 0.5 22
Appropriate priorities, Felicitous phrasing, Patriotic pricing, Track record, Laptrap, Turbo Prolog, AI Expert.
RELEASE 1.5 23
Ingram: Semantic error.
CALENDAR 24
Plan ahead!

Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016, (212) 503-5500

80 percent of the use. Moreover, although the sets of instructions for commercial and for scientific data processing diverged at the rarely-used end, both kinds of users had the most frequent 20 percent in common. In other words, all those fancy scientific instructions weren't used all that much.

Interesting results, these. This popular "new" RISC technology is just a throwback to the old days. The very first computers used RISC technology -- or at least, current ones use complex instruction set technology.

Back-to-basics is the thesis underlying RISC. But RISC applies only to the microprocessor. The implementation of a RISC machine includes memory management, compilers and operating systems, bus architecture, product positioning, etc. If you're going to take complexity -- or at least breadth -- out of the chip, you're going to have to add it back somewhere...

1 instruction + 1 instruction = 1 instruction?

In brief, performance depends not just on instructions per second (or MIPS, millions of instructions per second), but also on the number of instructions required to perform a given task. Complex instructions that would take a single cycle on a traditional, complex-instruction system may take several cycles of instructions on a RISC machine. The issue is the relative value of the improvement in MIPS versus the degradation in instructions required per useful unit of work. In other words, the cycle-time performance improvements of RISC are offset somewhat by the greater number of instructions and the increased complexity of the software it requires. For most practical purposes, the trade-off works strongly in favor of RISC -- a 2x improvement in MIPS typically is offset only 15 percent by a rise in the average number of instructions required, for example. (These figures are representative, but vary widely depending on the benchmarks -- or real-world problems -- under consideration.)

Impure power

performance = cycle speed times number of cycles needed

Thus the mostly self-serving claims that MIPS alone is irrelevant are true, although MIPS remains one of several interdependent factors to consider. To steal an analogy from HP's Joel Birnbaum, MIPS is revolutions of an engine, while performance is speed towards your destination. "Performance," always a fuzzy concept, depends not just on processing speed, but on I/O speeds, compiler performance, etc., in conjunction with the specific problem being run. The goal then is not to be a RISC purist, but rather to get performance as high as as possible in the face of other constraints. A RISC bigot would refuse to recognize any multi-cycle instructions, but out of the RT PC's 118 instructions, only 84 are single-cycle. The rest are commonly used two-cycle instructions.

The RT PC: Some technical considerations

As competitors hasten to point out, the RT PC bogs down on floating-point processing, and it lacks much networking support. Worse, it doesn't run much faster than a MIPS or two, about the same range as the significantly cheaper Sun 3. But why should it? While promising continued supremacy on the basis of specs, a Sun engineer concedes that "this is the machine we'll

be measured against" from now on. IBM as yet has neither need nor desire to sell more of the system than it can make, and production will be limited for the next year or so. Meanwhile, the RT's performance can easily be boosted. For starters, the basic chip is built with 2-micron NMOS; 1-micron CMOS, just about to emerge from industry labs, should be twice as fast in many areas. Compiler improvements could also speed up the RT by a factor of two.

Besides, as noted above, the system has significant performance advantages that are not indicated by MIPS. The salient features depend not on the chip or its speed but on the RT's overall architecture, especially its bus configuration, and on its memory management unit.

The 24 mbps processor-memory bus is separate from the RT's I/O bus, which itself is compatible with but higher-performance than that of the PC AT. (Such two-bus architecture is standard on workstations such as those from Sun and Apollo, which has just announced an AT-compatible I/O bus. The same two-bus architecture will no doubt be used in any system incorporating the Intel 80386, notably IBM ESD's next generation of the PC AT.)

Two implications: First, the RT PC can run faster, because processor/memory interaction isn't slowed down by I/O, and I/O itself is faster than on the AT. Second, the system can easily fit into the existing PC AT environment, sharing peripherals and even absorbing a PC AT coprocessor with little disruption. (The RT's software handles the interaction between the two, while the PC AT operates oblivious of the RT's presence.) Although there are some performance penalties in the use of the AT bus, the convenience of attachment to the PC world far outweighs them in the marketplace.

More impressive -- and unique -- is the system's memory management unit, a separate -- and complex -- chip that handles the system's 40-bit address capabilities. The RT can address a trillion bytes of memory directly, far more than most in its class. In addition to that sheer volume, the system boasts unusually fast access times because of an inverted-page structure that avoids the complexities and overhead of multi-level pointer-based systems.

Furthermore, because of data-tagging in the cpu, the RT can send out requests for up to 32 memory references (data or instructions) in succession without waiting for the first one to come back. Each response is identified (tagged) and stored in its own register or buffer and rapidly executed in sequence as soon as the system is ready. (This data-tagging, which operates at a low level to identify which data is in response to which request, is not the same as "tagged architecture," or hardware data-typing, which operates on a higher level to identify data *types* and speeds the performance of symbolic processing systems. The RT PC is well-suited for AI work because of its large, fast-access memory capacity, but it still lacks, as do Sun and Apollo, the tagged architecture found in standard LISP machines.)

Something old, something new; something borrowed, something Blue

While RISC technology came out of IBM's research labs in Yorktown (home of the 801 prototype), the RT PC comes from an independent business unit in Austin, Engineering Systems Products. Group director of the effort is Frank King (see Release 1.0, February 16), a chief designer of IBM's SQL data base query language. The actual system design work was overseen by Glenn Henry,

an IBM Fellow. IBM Fellows are rare (44 in total), and most are kept safely in research. Aside from being one of only a handful of Fellows actively involved in product development, Henry brings an interesting perspective to bear on the RT PC: He earlier managed the development and launch of the System 38. While the System 38 is probably one of IBM's finest systems from a technical point of view, it lacked software and third-party support at its announcement, and never fully recovered.

Not so with the RT. IBM has garnered wide support from developers, some of whom were paid for their efforts. Moreover, the system supports UNIX -- the least common denominator of the scientific-workstation marketplace.

So far, UNIX, even System V UNIX, is a fairly weak standard: It leaves a lot of specific design decisions open to the whims of the implementer. IBM's AIX, by contrast, uses a UNIX V foundation to support an array of incipient "standards" for interface, data management, data base management (Oracle/SQL), and the like. (Austin went outside to Oracle to get an implementation of its own dbms standard, SQL, we assume to avoid standing in line for the corporate development group to do a port of DB2.) AIX, which comprises about as much IBM code as AT&T code, offers character sets and fonts, pop-up menus, a soft keyboard, mouse support, multiple virtual terminals (which allow context-switching but leave the implementation of windows to developers or users), and a host of single-user, "friendly"-type features absent in vanilla UNIX.

That's the would-be standard stuff -- and precisely what the market has been waiting for: a set of tightly defined specs that they can grumble about and support for absence of anything compellingly better. (Give me a standard; any standard.) Current history also encourages quick support of the RT PC; Computervision and other CAD/CAM/CAE companies late in porting their software to the PC AT have suffered for that oversight.

On the proprietary side, AIX supports (as it must) the RT's proprietary memory management scheme that gives the system so much of its power.

Filling holes; drilling holes

IBM's RT PC, like the PC itself, is being launched experimentally. Observers have noted that its positioning is fuzzy. Precisely so. That is generally the case when IBM launches a new product into a market already served by an existing IBM entry. Although the RT fills a significant hole in the company's product line -- the scientific workstation -- it also encroaches on the departmental-processor market served by the System/36 (to say nothing of the territory staked out by the PC family). The System/36, a tired, low-performance multi-user workstation, clearly isn't up to the task (just as HP's 3000 series is running out of horsepower; see below). By offering underlying "RISC technology," however, IBM can dish up something that sounds more impressive than JAUWS -- Just Another UNIX Workstation -- but that fits the specs perfectly. It also enables IBM to have something proprietary -- or at least not immediately copiable -- while at the same time adapting easily to that partially defined standard-in-waiting, UNIX.

Is IBM committed to the RT PC? That depends in part on customers' interest in the system. The signs so so far are good: The IBM announcement at New York headquarters called it a "strategic family" and a new "direction" --

"those are words people in IBM kill for," one of them tells us. IBM has paid certain third parties to write software for the RT, a sign of *its* commitment; others are doing so for free, a sign of *theirs*. Approximately 2000 RTs are out in the field at developer sites.

Ultimately, we see the RT PC as a Trojan horse for hundreds of AI troops, invading end-user desktops and infiltrating the real world. "AI" applications can work in C on a PC AT, and real-world applications can incorporate little expert systems, heuristics, or other "AI" components, but performance will increasingly become an issue as these applications become more powerful (just as Lotus requires more machine than did VisiCalc). The 80386 can help to solve that in part, but the more flexible, potentially more powerful RT PC also looks like a superb candidate for that role. It addresses 40 megabytes of memory now; it runs at close to 2 MIPS now; it will have a number of high-visibility expert system tools -- with full support from their vendors -- within a few months; it runs LISP extremely fast, by all accounts. IBM's 80386, by contrast, likely won't even appear until 1987.

To be sure, the idea of putting AI on the end-user's desktop is not new; indeed, Xerox sells such a machine, the 1185, which runs both LISP and MS-DOS. But as usual, what's outside matters as much as what's inside -- and the RT has both.

HP's end-life kicker

The problems HP is addressing with RISC technology are quite different from IBM's. Indeed, IBM didn't have problems so much as opportunities. It sent some hardy souls off to start a new Independent Business Unit, much as it did for the IBM PC. If their project, code-named Sailboat, amounted to something -- great. If not, well, that was the purpose of an IBU -- to test out ideas without disrupting the mainstream of the corporation.

For Hewlett-Packard, by contrast, RISC is the solution to a debilitating problem: Users love HP's equipment, especially the 3000, but it's reaching the limits of its upgrade capabilities. It just can't be scaled to go any faster with its current technology. Either Spectrum works when the first machines reach the outside world later this year, and lets HP compete on performance as well as on the HP-based applications that its users know and love, or HP is in serious trouble as a high-end mini company.

Compare & contrast

HP faced a task not just of greater necessity but also of substantially greater magnitude than IBM's. IBM merely had to meet a loosely defined spec (for UNIX system V) that didn't correspond to any machine in particular; HP had to emulate a thoroughly defined system that had been in use, and solidified, over 12 years. On the other hand, HP had to achieve only a level of performance that was not a competitive *dis*advantage, since its existing customers were inclined to stick with the 3000 as long as poor performance didn't drive (or others' superior performance didn't lure) them away.

IBM faced similar problems with the System/36, but that product hasn't won the widespread following of HP's 3000 series, and is less vital to the company as a whole. So there was little incentive for IBM to build a RISC system that could run S/36 software. Besides, because of IBM's marketplace

power, third-party vendors are already endeavoring to port their S/36 software to the RT PC.

The different dynamics guiding HP and IBM are intriguing: Operating in a commodity UNIX market, most companies must stretch for performance. IBM has avoided this partly by the sheer force of its name, partly through AIX, a slightly proprietary version of the UNIX standard -- which the force of its name and the tailwinds of third-party support will help make a standard.

Hewlett-Packard, on the other hand, sees RISC technology as a way to help it keep its customer base and penetrate it more deeply. Customers who already have the 3000 will have no impetus to change. Those who wish to purchase turnkey systems built around the 3000 -- notably Ask's ManMan -- will not be deterred by performance issues (although the availability of ManMan on DEC equipment muddies *those* waters). But those who aren't using HP equipment already won't find any compelling reasons to do so.

Release 1.0 is published at least 15 times a year by EDventure Holdings, One Park Avenue, New York, NY 10016. Editor & publisher: Esther Dyson; associate publisher: Sylvia Franklin. Telephone: (212) 503-5500. *Copyright 1986, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced in any form without express permission.*

FACTS AS ARTIFACTS

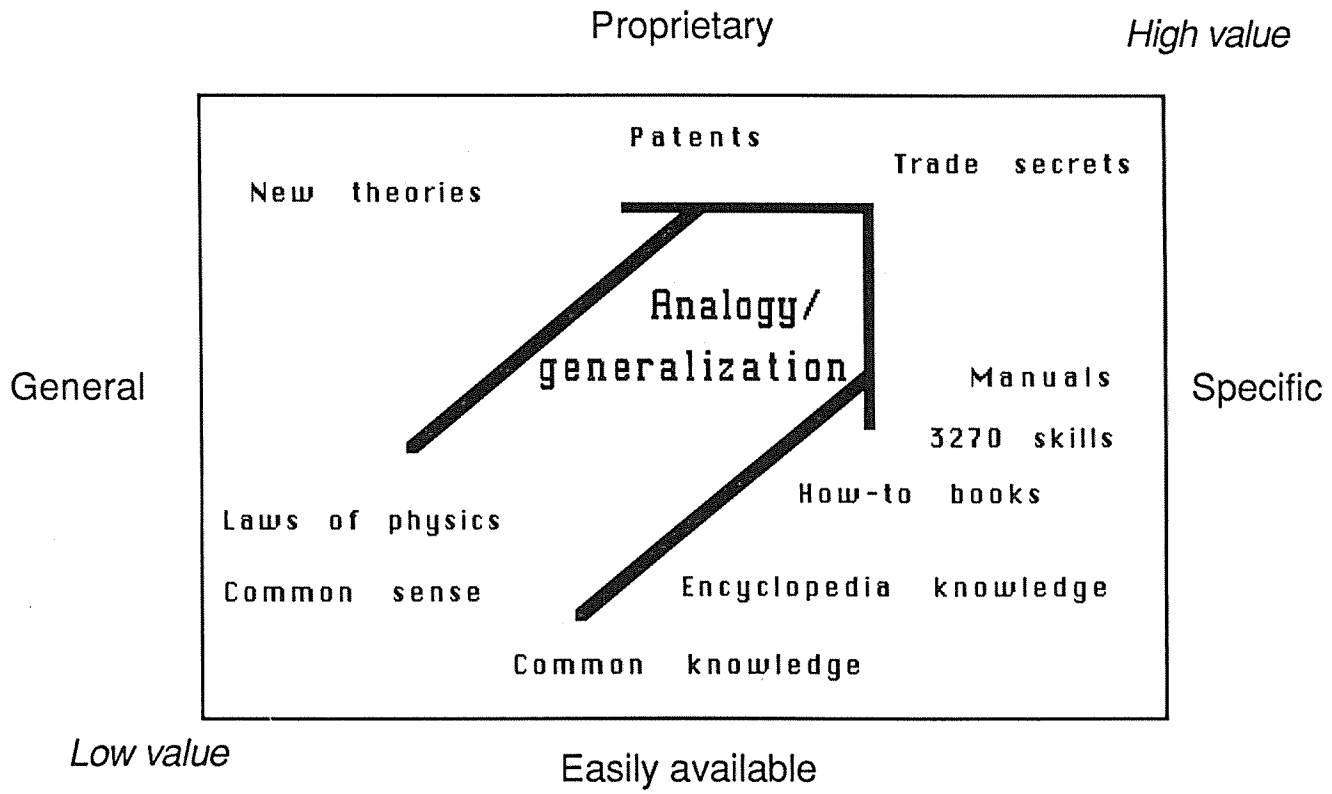
One way to describe knowledge is fuzzily, as imprecise coordinates on a graph. Let's consider some common types of knowledge. The terms we use are ours alone, and have no official or technical sanction. (Next month we'll have a clearer vision, but that's why we call the newsletter Release 1.0.)

- **Common knowledge:** What mothers are, what chairs are for, who's president, when (not *why*) water freezes, how long milk lasts on the counter and how long in the fridge, what happens when you lock a door, the differences between books and magazines, whether a particular name generally refers to a man or a woman or a dog, and so on.

- **Encyclopedia knowledge:** General facts about the world, how things work, geography, history, demographics, why water freezes.

- **Empirical knowledge:** Information found in manuals, such as how to build or repair specific things such as cars, washing machines, computers, Boeing 747s, power plants; street maps; mailing lists; how to read or even program a 3270 screen; etc.

A framework for knowledge



- **Frontier knowledge:** Speculations about cures for diseases, optimal circuit design, valuable market research and insights. Frontier knowledge can be general, as in new scientific theories, or the less-

solid edges of "empirical" knowledge. Frontier knowledge lies along the top of the chart above, because it is not generally available, or even believed.

- **Common sense:** What to do in unexpected situations; what things mean; why people do what they do; the reasoning equivalent of common knowledge, far less tangible.

The chart above positions assorted kinds of knowledge in a general framework (for another prespective, see across). Note that knowledge becomes more valuable as it is applied to specific instances and restricted in availability; as shown by the arrow, the application of analogy or generalization to general, public-domain knowledge raises its value. Empirical knowledge covers a large part of the grid; much of the more specific empirical knowledge qualifies as trade secrets or may be patentable or copyrightable. And remember, knowledge can slide around in this universe, depending on circumstances. Knowledge widely dispersed in the U.S. -- certain agricultural practices, for example -- might be valuable, proprietary information in Africa (not that its worth will always be recognized). The laws of physics, for another example, could be extremely valuable as they are applied to the production of energy. But sorry, fire is no longer a trade secret! Over time, trade secrets -- such as how to bake a cookie that's crisp on the outside, chewy on the inside -- become common knowledge. (Notice how we're expecting you to generalize and draw analogies from the examples we're providing here.)

Analogy and generalization

Clearly, the ability to render general knowledge specific increases its value, as does the ability to apply knowledge in a large number of situations. This ability comprises the ability to generalize or analogize. Generalizing we define as applying the same rules to items in a group; it's hierarchical: What applies to all members of a species also applies to all subspecies. Generalization may work top-down, as just noted, or up and down: Some facts true of cats are true of all housepets and thus are true of lapdogs. (Obviously, there's need to distinguish between things that are generalizable and those that are not.) Analogy is even more interesting: That's the ability to apply -- correctly -- the same rules to things or situations that are *not* part of a group (except as defined by that analogy). *Generalization and analogy tend to increase the value of general knowledge by applying it to specific things, and the value of specific knowledge by raising the number of things to which it applies.*

The intrinsic value of a unit of knowledge (however that may be defined) is a function both of its single-use value and the number of occasions in which it can be applied. But knowledge also loses value if it's easily available. Part of the trick lies in applying easily available knowledge to new situations -- working by analogy, to some extent. A manager/marketer from PepsiCo, for example, may have greater value in a market where his skills (i.e. knowledge) are uncommon.

Common sense

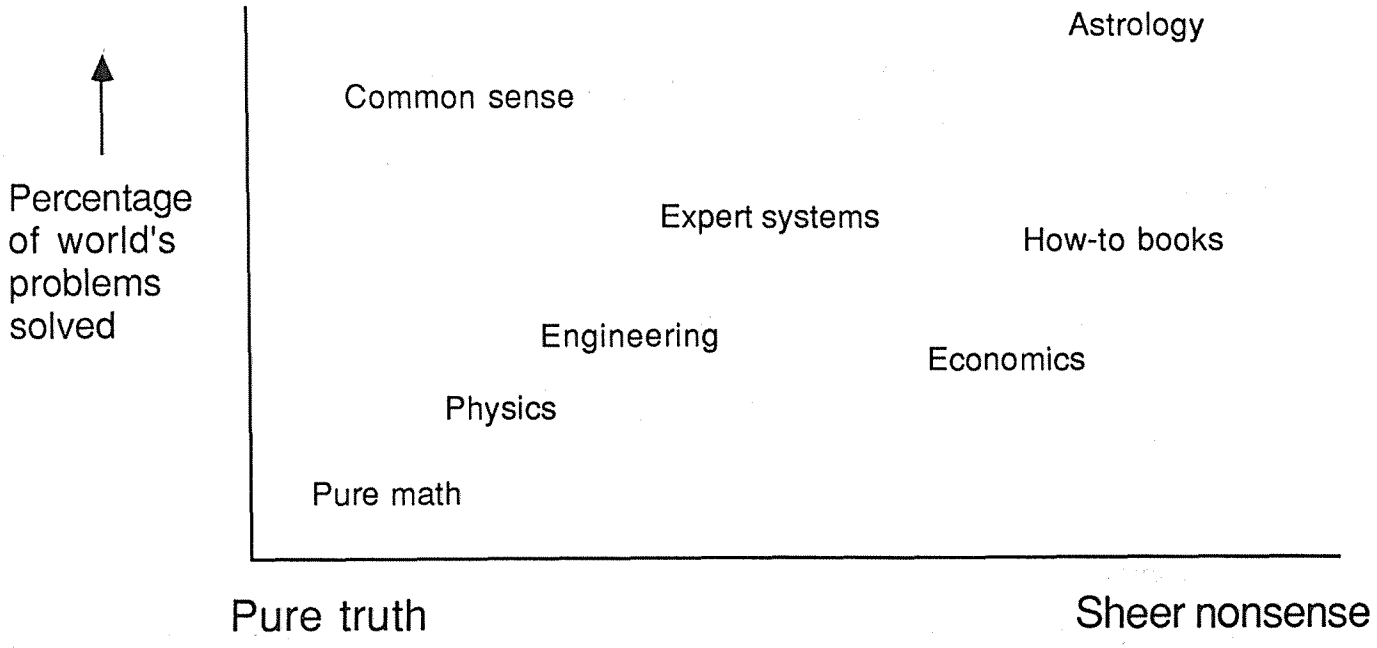
Common knowledge and common sense are akin to water, in that they are tremendously valuable and necessary, but generally so prevalent that their eco-

nomic value is low. By contrast, frontier knowledge may have potentially high economic value, but its potential applications may be few.

But what *is* common sense? Is it lots of general knowledge applied specifically? A certain kind of reasoning? It's lots of knowledge about lots of things, plus rules of thumb. Indeed, it's pretty much all the knowledge that's *not* in the encyclopedia, because it's so "common." For example, it's more likely for someone to be lying than for the laws of physics to be untrue. It's more likely for a clock to be broken than for a plane trip from San Jose to Seattle to take half an hour. And it's more likely for an employee to oversleep than for his clock to be broken every Monday morning.

Common sense is the ability to know when the general rules don't apply... error correction. Very simply, it's not writing down H-E-L-P when the answer to "Who's calling? is "Help!!" It's the ability to find alternate ways of reaching a solution, and understanding when an answer is not appropriate. It's the ability to avoid sending bills for $00.00 due April 13, but on a grander scale. Is it represented as rules? Analogies? Is it easy to represent, but so broad as to take impossible amounts of computer resources? Is it lots of rules, or just a few smart ones generally applied? Is it just the simple rule, "These rules don't always apply," with hundreds of thousands of variants? Is it those rules applied to common knowledge and encyclopedia knowledge, or is it a special kind of knowledge of its own?

AI folklore



(Now where do we put Ann Landers?)

THE OMNIVOROUS MCChine

Next month we're going to talk about a system that replaces a 3270 operator: It doesn't do much, but it can talk over the phone to a limited extent, push (virtual) buttons, and read the answers off a (virtual) screen back to the caller at the other end of the phone line. Its economic utility is potentially enormous -- $20,000 annually times millions of such operators. And the machines never get sick, they work on weekends when bank-balance inquiries and responses to steak-knife commercials are most frequent, and they don't require benefits, management, vacations, etc.

But few would ever consider such a machine "smart. Who cares? Generally, we don't. Whether or not, or how much, something has "real AI" doesn't faze us in assessing a commercial product. Yet the concept of a machine that does appear to be smart, that can apply common sense, that wouldn't be stymied by a non-3270 crt, certainly appeals to us intellectually...

So it does to Doug Lenat, a former assistant professor from Stanford whose request for tenure caused sharp divisions among the faculty there. He has now found shelter and funding as Principal Scientist at Austin's Microelectronics and Computer Technology Center -- MCC, the research organization funded by 21 computer/electronics companies. Lenat plans to take 10 to 20 years to realize his vision of a really smart computer, one with common sense. This may seem a while, but consider how long it takes to grow a 3270 operator -- of the kind that can go home after work and fix dinner or take in a movie. Other people -- as far back as John McCarthy -- have tried similar things before, but Lenat is convinced he has better equipment and better methods of representing analogy/generalization that give him a higher chance of succeeding than his predecessors. Success or no, he's sure to discover *something* that could apply to less ambitious commercial efforts.

Nine months...twelve years

Lenat argues that all the expert systems wandering around are simply demo systems: They solve only the specific, limited problems they're custom-tailored to solve. More interesting would be a system that could figure out by itself how to solve a much wider range of problems. These are the common-sense capabilities Lenat is trying to build into CYC -- a homonym for "psych" and a contraction of "encyclopedia."

Consider the edict that has governed education over the past few decades: Don't teach children facts; teach them how to *find* facts. They can always look in the encyclopedia. This approach may work, but it's amazing how many facts children pick up on the way. Specific knowledge is always most valuable, but the overall rewards for having it may be low because each "piece" it is so rarely used. Better to have a generalized system that can handle anything. CYC aims to be a combination of these -- rich with massive amounts of specific knowledge, and also in some guiding "common sense."

Learning curve

The underlying thesis is the existence of a learning curve for knowledge acquisition: A twelve-year-old may appear smarter than a two-year-old not because his brain cells are any better, but because he knows so much that he's better equipped to learn more. Consider, say, a new software program.

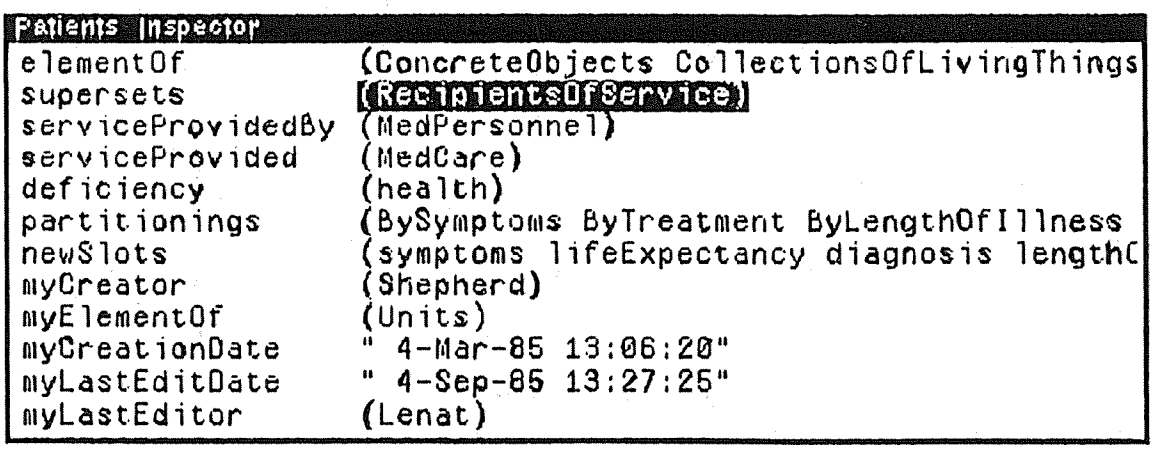
At first it may seem completely foreign, but once you grasp part of it, it's easier to keep on exploring new features. Or consider the concept of software in general. Once you learned VisiCalc, it was easier to figure out how to use a word-processor. The concepts of storing, marking and transforming data applied across the board, albeit in different implementations.

Lenat is trying to create roughly a twelve-year-old -- intelligent, aware, but without the adolescent propensity for soul- and self-searching. It will have all the knowledge Lenat and his team can transfer from an encyclopedia, plus that much again of everyday facts and common-sense rules. All told, it should take up about 200 megabytes, Lenat estimates, two orders of magnitude more than current commercial expert systems.

How to skin a cat

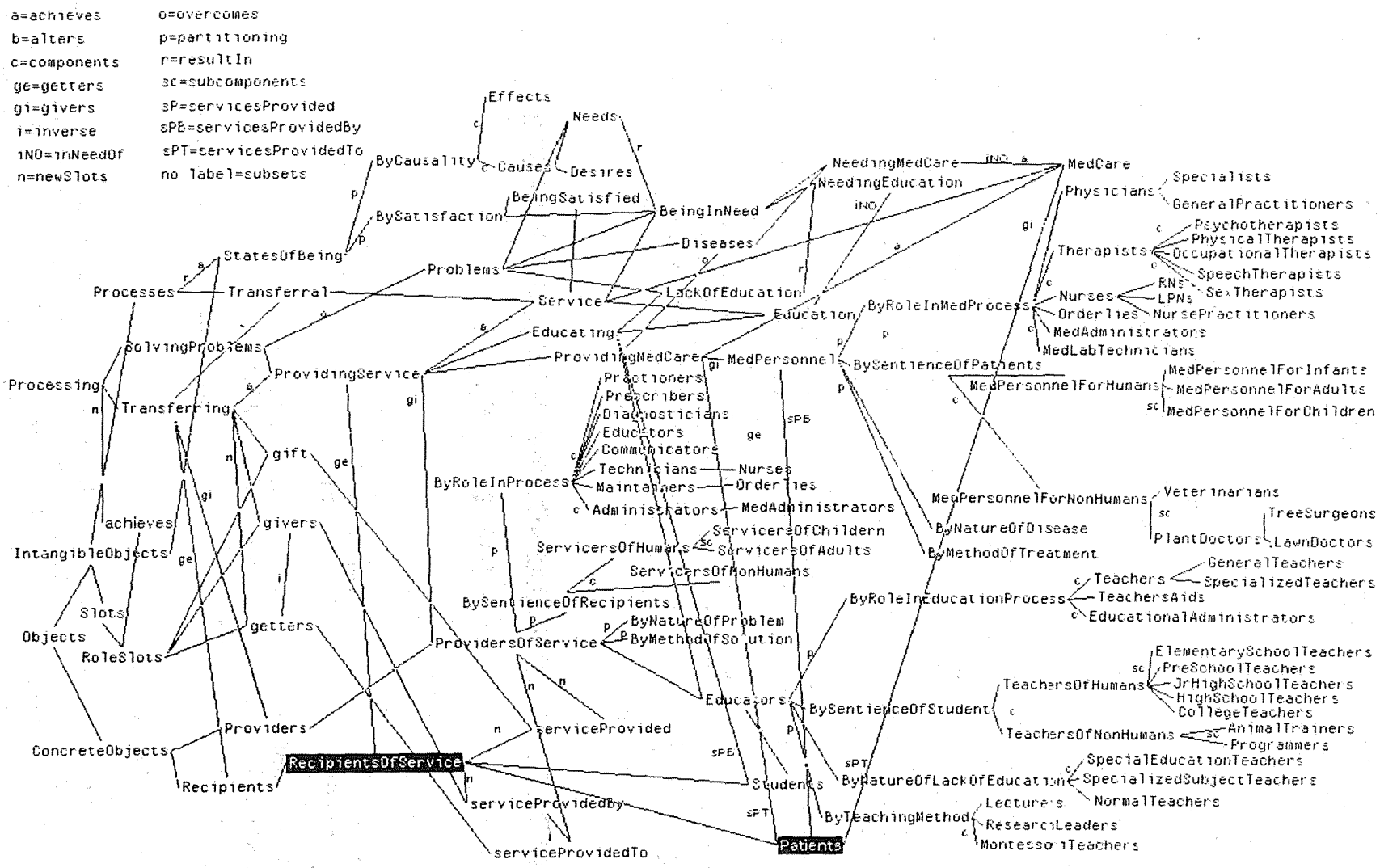
The "traditional" expert system, modeled on Emycin, consists of a bunch of rules, which are blindly executed to reach some kind of "goal" or conclusion. A slight improvement on this embodies some control, directing which rules to test first and optimizing that order to some extent. Control uses rules about rules: in what order to test rules, how to go about searching for answers to different kinds of questions, even what kinds of questions to ask to elicit further information. These are sometimes called meta-rules.

Representing all knowledge just as rules, however, gets to be extremely cumbersome. Far better to represent it also as tables; relationships; "frames" -- sets of attributes common to several members of a group; "scripts" -- sequences of events that define an action (transportation) or experience (restaurant-going); or even embedded formulas or cross-application subroutines that can go out into other data bases or otherwise get information from the real world by querying a user at a terminal, reading a sensor, or running another application to obtain a result.



Representing knowledge

One of the most convenient -- and the most easily comprehensible -- forms of knowledge representation is frames; see the frame above, which shows facts about patients. Frames are a familiar concept: Patients are concrete objects, and living things, larger groups with which they share some but not


a=achieves
b=alters
c=components
ge=getters
gi=givers
i=inverse
iNO=inNeedOf
n=newSlots
o=overcomes
p=partitioning
r=resultIn
sc=subcomponents
sP=servicesProvided
sPB=servicesProvidedBy
sPT=servicesProvidedTo
no label=subsets
Effects
Needs
Causes
Desires
ByCausality
BySatisfaction
BeingSatisfied
BeingInNeed
NeedingMedCare
NeedingEducation
MedCare
Specialists
Physicians
GeneralPractitioners
Psychotherapists
PhysicalTherapists
OccupationalTherapists
Therapists
SpeechTherapists
SexTherapists
RNs
LPNs
Nurses
NursePractitioners
Orderlies
MedAdministrators
MedLabTechnicians
StatesOfBeing
Diseases
Problems
Processes
Transferral
Service
LackOfEducation
Education
ByRoleInMedProcess
SolvingProblems
Educating
ProvidingMedCare
MedPersonnel
ProvidingService
Processing
Transferring
Practioners
Prescribers
Diagnosticians
Educators
Communicators
Technicians
Maintainers
Administrators
Nurses
Orderlies
MedAdministrators
ByRoleInProcess
BySentienceOfPatients
MedPersonnelForHumans
MedPersonnelForInfants
MedPersonnelForAdults
MedPersonnelForChildren
gift
achieves
givers
IntangibleObjects
ServicersOfChildern
ServicersOfAdults
ServicersOfHumans
ServicersOfNonHumans
MedPersonnelForNonHumans
Veterinarians
TreeSurgeons
PlantDoctors
LawnDoctors
ByNatureOfDisease
ByMethodOfTreatment
BySentienceOfRecipients
Slots
getters
Objects
RoleSlots
ByNatureOfProblem
ByMethodOfSolution
ProvidersOfService
ByRoleInEducationProcess
Teachers
GeneralTeachers
SpecializedTeachers
TeachersAids
EducationalAdministrators
ElementarySchoolTeachers
PreSchoolTeachers
JrHighSchoolTeachers
HighSchoolTeachers
CollegeTeachers
TeachersOfHumans
Educators
BySentienceOfStudent
TeachersOfNonHumans
AnimalTrainers
Programmers
Providers
serviceProvided
ConcreteObjects
RecipientsOfService
SpecialEducationTeachers
SpecializedSubjectTeachers
NormalTeachers
ByNatureOfLackOfEducation
Students
Recipients
serviceProvidedBy
Lecturers
ResearchLeaders
MontessoriTeachers
ByTeachingMethod
serviceProvidedTo
Patients

all traits. Patients are also recipients of service, and in that capacity they are related to medical care and to medical personnel. They have attributes: life expectancy, symptoms, etc. A single thing is generally part of several hierarchies. When you try to tie all these hierarchies together, it takes a lot more than just a relational data base. Forget concepts; just the sheer number of tangible things in the world creates an overwhelming number of relationships, as shown opposite.

Thus CYC can deal with more than just static things. Medical care has an effect (you hope) on its recipients. What about a train? Or a restaurant? The concept of either includes a flow of actions: Transportation, destination, a change in the location of the items that have "carrying" or "riding" links to the train. A restaurant involves a series of actions, but they vary: In McDonald's you pay first, then eat. At Lutece, you eat first, then pay (or wash dishes!). These sequences can be represented in scripts, or in special kinds of relationships among things.

The soul of CYC

All of these are standard knowledge-representation concepts. What's relatively new, or newly implemented at least, is CYC's predilection/ability for analogy. Analogy is a powerful tool. On an ad hoc basis, it can make the relationships between objects and concepts explicit when the system is looking at them, obviating the need for those relationships to be built-in for each case, thus saving temendous resources. Analogy can also can define the structure of frames for new concepts. It can lead to new ways of finding answers. It can indicate relationships between new objects. It can even suggest metaphors and other relatively high-level "thought" constructs that make a machine look "intelligent" -- now that we're used to expert systems and no longer consider a diagnostic system intelligent.

Analogy is a sort of pattern-recognition capability that matches patterns of slots within frames rather than the precise data in the slots. Moreover, it recognizes patterns that aren't necessarily clearly defined. CYC can adjust its "fuzziness" -- or tolerance of inexact matches -- according to circumstances. It all depends on how imaginative you want the system to be.

Analogy/generalization should enable CYC to appear far smarter than its 200-MB size would indicate. Most of what it "knows" will be generated on the fly: General knowledge of car-driving and map-reading, for example, enables Juan to get to Alice's house even though he's never driven that particular rental car or been in Alice's hometown before. Indeed, most "knowledge" is the application of generalities to specific situations. Most objects and siutations we deal with are simply specific cases of more general concepts. The relationships among them, likewise, reflect general relationships among classes of things. Once you know about food, you can figure out what to do with a hamburger -- or a rattlesnake, if you're so inclined.

Take, for example, "Marketing is War." Given that, and a few examples -- such as Sales forces are troops; Market share is territory; Competitors are enemies; New products are new weapons -- the system may start to look for appropriate analogies itself. What is nuclear warfare, for example? Irrational price cutting? The use of antitrust? What are third-party vendors -- allies, or advance routes for troops/products? And then there are rules that apply to both: For example, don't divide your forces. It's

easier to defend territory than to capture it. These rules could be kept general, and then applied to a third situation -- medical treatment, for example, with opposing forces of antibodies and bacteria...

"Most analogies 'work' because there is common causality," writes Lenat. "Analogies are *useful* because often that causality has resulted in many more shared attributes that haven't yet been noticed." (Take Al Ries's recent speech and book on Marketing Warfare as an excellent example; Release 1.0, February 16.) What this means is that marketing and warfare may share many slots -- and that knowledge about one suggests knowledge (real or missing) about analogous attributes of the other.

In the beginning there was data entry...

The fundamental cost of a knowledge system is not its hardware or software but the sheer effort of getting the knowledge -- facts and relationships -- in there. Thus these newer, richer methods of knowledge representation constitute a significant advance. (Note that a table *can* be expressed as a series of rules or declarative statements, or vice versa, but it's awkward.)

Each new piece of knowledge reqires the implicit creation of hundreds of such links, represented by slots in frames which either state a specific link or imply many such links borrowed from a "parent." The links are actaully created only ad hoc, when a specific item or situation is being examined by CYC.

The effort that will be applied in teaching CYC, the actual acquisition of knowledge, is awesome, but these borrowed links and frames greatly ease the burden. Ultimately, the hope is, the task will get easier. Most of the frame templates and links will already exist, so that each new piece of knowledge can be readily assimilated. Objects will have familiar characteristics, and relationships will already be defined...

This desirable state of affairs is rendered tangible in time-saving data-entry routine of "copy&edit" -- i.e., a new piece of knowledge is defined in terms of how it resembles and differs from another, existing peice of knowledge. The growing use of "copy&edit" will reflect the system's increasing ability to "learn" as it incorporates a wider range of frames, scripts and rules to copy&edit from.

Well of wisdom?

At the moment, CYC is a sparse, inert webbing of facts, built with the aid of eight Symbolics LISP machines (replacing six Xerox LISP machines). All these facts exist somewhere within a multi-dimensional space, of which one can see small slices on a beta-version color graphics terminal from Symbolics. Two people now, 24 by next year-end, spend eight or more hours a day feeding CYC information from an encyclopedia.

Can the sheer knowledge CYC will embody amount to wisdom or common sense? The answer really isn't known. It's fairly clear to anyone dealing with today's expert systems -- incorporating at best a few thousand rules and facts about limited domains -- that these systems don't approach wisdom at all. You can pretty much "watch" them think, as they blindly try out

hypotheses and come up with reasonable answers, operating within the parameters they were given at the outset. Is the limitation the machines, or the knowledge they possess? Will an overwhelming quantity of knowledge result in a change in *quality*? Is the flash of insight of the "creative mind" no more than the ability to see analogies?

The second question -- not necessary to answer yet -- of what value will the system be? Lenat's general hope is that it will enable the creation of machines (expert systems is too limiting) that display common sense, i.e. that are far more adaptable than anything commercially envisioned now. Of course, the enticing thing about Lenat's project is that if he creates only one, it can be replicated at low cost -- far lower than twelve years' worth of diapers, schooling, hamburgers, and overalls -- to say nothing of parental wear-and-tear. Moreover, the hope is also that once the system gets past a still uncertain "critical mass" of knowledge, it will be easier to teach -- or may even, with some future natural-language capabilities, teach itself.

And finally, how much will we *want* it to know? Linked up to the outside world, it could theoretically subsume all the electronic knowledge that exists. Frightening thought. Whose goals and priorities would the system ultimately reflect? If not its own (what are those?), then whose? Doug Lenat's? The government's? Those of a committee of 21 companies?

FRED'S PARTS & PUMPS

Suppose you want CYC to give Fred some advice on which computer to purchase for his gas station. Here we come to Lenat's set-piece, an example he recently worked through for his shareholders, the 21 companies who fund MCC, at a recent progress review. (Lenat finds the mandated monthly reports "a little fine-grained," given that his progress, like a child's growth, is best measured yearly. The last few months, for example, have been devoted mostly to porting the system from Xerox machines to Symbolics machines, with a few improvements in the system's underlying logic along the way.)

The Fred's-computer problem -- which isn't totally implemented yet, but we're not talking about a commercial system here -- gives some idea of how CYC attacks (or will attack) a problem. Most expert systems get fairly explicit instructions, and run a problem. What is the best investment for a yuppie...? ("Goal = BestInvestment." The system finds the rules that give a conclusion for BestInvestment, finds the rules that lead to those premises, and chains backward, interacting with the user, until it comes to the conclusion. Or it can chain forward, depending on how it's constructed.) But CYC can handle a problem any number of ways, *and* figure out which is most appropriate. Consider again this question: What kind of computer system should Fred buy for his gas station? CYC knows of at least four approaches to the problem:

1. Find out his requirements and find a computer to meet them.
2. Work back from the choices available and maximize results.
3. Analogize from what other gas stations are buying.

4. Analogize from what else Fred buys (i.e., always the cheapest, always IBM, always whatever his brother buys).

The meta-rules in the system can determine which it tries first, and how hard it works to find out Fred's requirements before deciding to try other methods. For example, if it doesn't find the slot "requirements" in the frame for Fred's Parts & Pumps, should it look at the frame for requirements and see if there's other information available from which it could *determine* Fred's requirements? For the purposes of this virtual demo, that's what it does, rather than check out what the other gas stations are buying. (Here's where parallel processing might come in handy, allowing a system to try several approaches simultaneously and then compare answers or, in a processing-intensive problem in a resource-short situation, let the first answer win.)

So, CYC chugs along and checks out what it knows about computer requirements, and finds out that they consist of speed and memory requirements, which can be computed from transaction rate and number of objects stored in memory. And so forth and so on. Ultimately, CYC figures out Fred's Pumps & Parts' transaction rate by dividing his yearly revenues (derived from its identification as a small business) by an average transaction of $15 -- derived from the cost of gasoline times the normal capacity of a car's gas tank. Obviously, there are slots and frames and relationships here that we haven't gone into, but they're all representable by the system.

What if one user calls Fred's Parts & Pumps a "gas station" and another user calls Fred's competitors "service stations"? That problem is generally avoided by the use of the pre-configured frames. A researcher filling in the frame for Fred's might type in "gas station." CYC would respond, "What is a gas station?" The researcher would then know to check the roster of types of small businesses to be sure that there wasn't a synonym already in the system, and would define that synonym. (CYC might also ultimately recognize the existence of a lot of similar frames in two separate categories and question its builders about that similarity, asking, What differentiates service stations from gas stations and from filling stations?)

To continue with the example, it turns out that Fred's actual speed requirements are so low that they're irrelevant (there's no computer that slow in production), but he does need a hard disk. The example ends there, but in use the system would next move on to check out Fred's buying patterns, if any. Finally, it might determine what the competition is buying and suggest Fred buy the same. How different is that from what a high-paid consultant would do?

This example shows quite clearly that you don't have to be very smart to be a computer consultant (of that ilk, anyway). But you need to have a lot of little bits of knowledge and know how to apply them.

For further information, refer to the excellent article by Lenat, Mayank Prakash, and Mary Shepherd in the Winter 1986 issue of AI Magazine, available from the AAAI at 445 Burgess Drive, Menlo Park, CA 94025-3496; (415) 328-3183.

ARTIFICIAL WEALTH

Prices of initial public offerings always look too rich. The truth is, most of them are. Chances are most newly public companies will stumble before they have a chance to reach the glowing projections that investment bankers whisper in investors' ears. ("Of course we're not allowed [by the SEC] to make projections, but (1) if current trends continue you could see... (2) the company is basing its budget on expectations of... (3) our assumptions indicate earnings of [whisper, whisper] per share by 1988.")

But there's always one in ten that will turn out to have been incredibly cheap. Forget Vector Graphic, Victor, CADO Systems, MicroPro, et al. Remember Tandem at $1.83, Cray at $1.10, Intel at 93 cents (all adjusted for stock splits). The hope, of course, is that the companies below will turn out to be that one in ten -- or those two in twenty.

Teknowledge shall make you rich?

We have rarely seen such scorn aimed at a public offering as that garnered by Teknowledge with its recent prospectus. Concern focused on the price, six times revenues of a mostly unprofitable company; a profusion of rich option plans; and reliance on revenues from "strategic investors." Nonetheless, the company successfully sold 2 million new shares of stock last week at $13 per share, 500,000 more than committed, and at the high end of the range of expected prices. (However, the price didn't jump afterwards, indicating it was what bankers call "fully priced.") All these details have been amply covered elsewhere. Now for the rest of the story:

What's frustrating about Teknowledge is that management says all the right things, but doesn't seem to understand their implications. After much discussion at a recent company presentation of the importance of integration with the real world, and of how the company achieves it by writing its tools in the C language, management proceeded to show a slide of the "artificial intelligence market." We fully expected chairman Lee Hecht to say, "...but of course this is irrelevant; we're in the overall software market." He didn't. Likewise, when asked against whom they assess themselves as a company, management listed not Cullinet and MSA, but rather IntelliCorp and "the rest of them [which] are private."

Products

Teknowledge's products do less and cost less than competitors'; the question is whether they do the right subset. Commercial systems built with Teknowledge's shell include a variety of diagnostic and repair systems, and other rule-intensive applications.

Teknowledge's products -- for the next year at least -- consist strictly of rule-based systems. They're thus very useful for rule-type problems, a distinct class of problems, while its competitors deal better with broader problems that require a wider variety of knowledge representation methods. By analogy, consider the difference between a spreadsheet and a data base: Like a rule-based expert system tool, a spreadsheet is optimized for certain kinds of problems, even though it can also be used to perform some data base functions. Neither product is intrinsically better; each is more useful in the appropriate context. But the analogy goes further: Spreadsheet sales

sparked the micro market, but ultimately the data base market (on networked micros) will be larger than that for spreadsheets. Like knowledge-representation-rich expert systems, data bases take longer to set up than spreadsheets, but they address a broader range of problems and are more integral to a company's operations.

Revenue sources

Although Teknowledge has not been terribly good at making money, its recent imbalance of consulting fees over product revenues has been due mostly to a wholesale conversion of its AI-language-based products into C, the better to infiltrate the normal world of commercial data processing. During this conversion, the company held back from selling its products, and suggested that customers wait for the C versions. When marketing starts up again, and the current roster of salespeople grows from only four people to 20 by the end of the year, product sales should presumably pick up. While consulting revenues aren't leverageable, as product sales are, the current state of the industry is such that products without support aren't much use either. One could do worse that to be an AI consulting company: The greatest shortage is of people, not of expert systems tools. But of course a consulting company should have little need for capital.

Perhaps the biggest uneasiness surrounding Teknowledge concerns its reliance on "strategic investors" for almost half its business. The answer to that one is that for the moment they represent a good source of business, and that the problem isn't their role as strategic investors, but rather the company's reliance on a handful of customers for most of its revenues. That problem should be alleviated if the company can sell its tools successfully.

While strategic-investor customers don't bother us (and we assume they're more likely to stick around than non-strategic non-investors), we *are* concerned at the concept of strategic investors in general, since they tend to fund a company without much caring whether it's profitable -- as illustrated by Teknowledge's retained earnings deficit of $9 million, which has been funded by $24 million in "start-up" financing over the last five years. But that problem should be alleviated by the current financing, which puts a quarter of the stock in the hands of the public. Whether pressure from those investors -- presumably interested in profits rather than more direct benefits from Teknowledge's activities -- will push the company into consistent profitability remains to be seen.

ORACLE ON A ROLL

Following an aborted attempt in 1983, Oracle Corp. of Belmont, CA (although it grew up on Sand Hill Road in Menlo Park), has now gone public. This is one of the richer offerings of the season, valuing the company at about $200 million, or six times revenues, and something like 80 times earnings per share. The offering sold out this week at $15 per share, and promptly rose to $21.

What investors are looking at instead of those numbers is the company's astounding ability to pick up OEM and corporate-standard business. Presumably for speed (months to implementation, not nanoseconds of performance), IBM itself went to Oracle to get the dbms for its System 88 (a box OEMed from Stratus) and, more significantly, for the RT PC (see page 1). Note that Oracle is basically a third-party version of IBM's own SQL/DS and DB2, although ported to a wide variety of machines and with the usual enhancements required to survive in the IBM-compatible business. Other OEMs include Xerox, Sperry, and Prime.

About half Oracle's revenues come from sales of the DEC VAX version, and much of the rest from IBM mainframes and PCs. A primary attraction is that you can run Oracle (or IBM's SQL/DS or DB2) on your mainframe, and trade applications or information seamlessly with a VAX, Apollo, Pyramid, Convergent Technologies or other machine also running Oracle. This ability has garnered Oracle a sizeable business with large end-user customers, including GTE, General Motors, GM itself, and numerous other boardroom names.

Oracle's only weakness is in the VAR market, where ostensible competitors Unify and Informix are flourishing (but generally with smaller chunks of business). Oracle has concentrated on friendly, end-user tools rather than the small-business application-building tools beloved of VARs. So far it seems to have by far the greener grass.

"We're only going public because we're doing business with Ford, GM, and AT&T, and they want us to be," says Ellison. But the balance sheet is also a little thin. Offsetting a healthy $5 million in retained earnigs, the books also carry $14 million in receivables, or close to the last half's revenues of $18 million. Write-offs, at least, have been minimal; Oracle's corporate and government customers are reliable payers, but slow.

Only half the stock is being sold by the company -- the rest is chairman Larry Ellison selling 100,000 shares (out of 4.6 million he owns and 13.4 million to be outstanding) and some early investors cashing out.

A VOLATILE LOOK AT CD-ROM

We recently attended Microsoft's conference on CD-ROM. As with any meeting of a self-help organization, the wonder is not in what you learn, but simply in the realization that you are not alone. That should be a spur both to the medium's credibility and to people's eagerness to get moving before the competition does. We formed some quick impressions -- mostly questions -- which will bear further examination later.

Although publishers and entertainment industry types were well represented on panels and in the audience, the overall content of the meeting was heavily computer-industry-oriented. Each of the three groups has a lot to contribute (and to earn for its efforts).

The biggest factional split -- which would merely be market segmentation in a group less convinced that *its* way is the only way -- was between those who see CD-ROM as a peripheral to a computer, and those who see it as an integral part of a dedicated player-cum-medium, or appliance. Each side offers its own, not-yet-fixed standard. Probably the entertainment types will miss some of the power that the computer types could provide with their data processing skills, while the computer types may not pay enough attention to issues of interface and intuitive interaction, and plain old visual and aural quality. And neither knows as well as the publishers that information by itself is valuable, and can be made far more so with the incorporation of accessibility and clarity.

In the business market, for the moment, CD-ROM is essentially just another storage medium, although one of substantially greater cost/performance than what we have now. It is also, significantly, an excellent distribution medium for publishers of data, and will ultimately change standard notions of what information any business should have immediately available.

Problems or opportunities

The value-added engendered by CD-ROM will be not just in the data, but also in its organization. The sheer volume of data that can be stored in CD-ROM requires and invites more powerful methods of indexing and retrieval than title, subject and keywords, and full-text search. (Powerful "indexing," in a sense, is what gives CYC its "intelligence;" see page 10.) Specifically, there's the opportunity to take "flat," linear text -- read from beginning to end -- and make it multi-dimensional. To understand the concept, consider the achievements of the outline processor, which makes text two-dimensional: You can read through it straight, or you can move up and down, from headlines to subheads to subpoints.

With appropriate software (for example, Xerox's NoteCards, Release 1.0, December 31, 1984), you could do considerably more than that. Imagine a document annotated with maps, tables, supplemental texts, all available to pop up at the push of a function key at the appropriate point marked in the text. Or imagine a document that began with a menu that let you select the point of view you desire: "Divorce proceeding #23901. Select (1) Juan's story; (2) Alice's story; (3) Correspondent's story; (4) Children's story." Or more seriously: "Current trends in CD-ROM. Select: (1) hardware angle; (2) software angle; (3) business market; (4) distribution issues." Some of the text would be common to all versions, but might appear in different

order and with different highlights in each, while each version would also have text specific to it left out of the other versions.

Peripheral or player

The peripheral or player/accessory or appliance issue is not an issue: It's a feature of the marketplace. We believe that CD-ROM players, likely using the CD-I (for CD-Interactive) standard propounded by Philips and Sony, could help revive the home com-- er, expand the home electronics market. If Apple were to launch its much-rumored 16-bit Apple *//x*, the combination of that with a CD-ROM peripheral (and a generous trade-in offer to current Apple // owners) might even revive the home *computer* market. The Mac is certainly a nice, friendly machine to front-end CD-ROM data bases, as shown in several demos at the conference, but the Apple // installed base and market positioning render it uniquely suited for the CD-ROM home market. (Atari and Commodore were barely visible in Seattle; Apple had several attendees.)

Active or passive

Do home users really want interactivity? If so, why do they watch so much TV? And if all they want is interaction, why not settle for video games and interactive software? The demos of encyclopedias and time lines were great, but how many people read encyclopedias or watch documentaries?

Application or format

"Applications!" sniffs Stan Cornyn of the Record Group, a longtime record-business executive turned CD-ROM entrepreneur. "What are the applications for television?" Touche. What TV has is not applications but formats. Formats are tougher to implement than applications: They're less tightly defined; it's harder to tell if they work, let alone *make* them work; they need more creativity; they need a lot more substance -- megabytes of text, images and sound, in fact. CD-ROM will dramatically raise the potential quality of graphics and sound over what is available now on video games or computers.

So what are the formats for CD-ROM? It's got to be more than encyclopedias, time lines and interactive self-help...

RELEASE 0.5

Appropriate priorities... (from a request for proposals for future AAAI conference sites:) "The proposal should include the following information: ... 5. Description of local regulations (e.g. labor union laws, liquor licenses..."

Felicitous phrasing... (from Bill Krause:) "'Strategic alliance' is what you call your first three big customers." Krause, president of 3Com, is soon to be president of Convergent; these two companies have made their strategic alliance tangible.

Patriotism in pricing... (from a Kodak document soliciting bids for pc software site licenses:) "It obviously is not in Kodak's nor our country's best interest to continue paying high per-copy prices for software products that are purchased frequently...."

Track record... If there's one thing venture capitalists like, it's a track record. Phil Cooper, who had started Computer Pictures in 1980 with $500 in funding (and $2 million a year later), sold it in 1982 to Cullinet for $14 million, giving his VCs a quick return. Thus Cooper had a relatively easy time raising $11 million for Palladian Software, maker of the Corporate Financial Advisor. That's doing so well he has time not just for aviation but for another startup, this time with an old pal from Cullinet, Charles Bachman, father of the network data base architecture which has become such products as IBM's IMS and Cullinet's IDMS data base management systems. The new company, called Bachman Information Systems and located next to Palladian in Cambridge, MA, is developing intelligent tools to assist in the design and implementation of IMS and IDMS applications. The system will run on a combination of LISP machines and real-world (IBM mainframe) hardware, using embedded knowledge specific to IMS and IDMS data dictionaries and applications. Cooper is a co-founder, and Bachman is ceo pro tem. More announcements will be forthcoming.

Bachman IS has raised more than $2 million from Venrock, Kleiner Perkins Caufield & Byers, Abingworth, and Cazenove (all four Palladian investors), as well as Palladian itself, and, intriguingly, Symbolics.

Laptrap... Zenith's recent win of the widely-rumored IRS bid for 15,000 laptop portables offers many lessons. First of all, quality does count -- and Zenith has regularly been racking up revenues on contracts where business is due to product analysis rather than marketing clout. Second, the episode illustrates just how easily speculation can become accepted wisdom. We suspect the rumor surfaced somewhere, was confirmed by people who had heard the rumor, and gradually fed on itself. In the same way, we feel, the laptop market itself may be the creation of journalists -- one of the few groups of people, in addition to the revenuers, who have much need for such a machine. As PCs become more ubiquitous, the need for laptops will diminish rather than grow. Just listen to Compaq, which had one -- close but no cigar -- two years ago, and abandoned the project. Product wasn't good enough, the company says. Or put it another way: The market doesn't always need the products that it is technically feasible to build.

Welcome! as Apple said to IBM... (from Arity Corporation, vendors of Prolog and associated development tools, on Borland's launch of Turbo Prolog:) "Without having seen Borland's product, we are very excited at the prospect of having a company like Borland put its weight behind a language that we have believed in all along. With Borland bringing up the low end (with a Prolog for everyone so to speak) we are counting on remaining in the high-end of the scale with a serious, sophisticated product."

Without having seen the product either, we find the claims for its performance extraordinary, but we remain open-minded. And yes, the market is young enough that Turbo Prolog probably *will* expand it, with salutary results all around. (That said, we still prefer LISP, although Prolog remains an excellent teaching tool -- the Pascal of AI, so to speak.)

AI Expert... The publishers of *Computer Language*, a successful magazine for serious hackers (i.e., they care as much about applications as about programming), will be launching *AI Expert* this summer. The magazine will be aimed at individual users of AI languages and tools, since advertising of those will help fund the venture. (Turbo Prolog should help expand *this* market too.) Still, we hope the editorial will also focus on how much you can do in "standard" environments. Adjustments will be made on the basis of feedback from the premiere issue. Since we're on the advisory board along with a bunch of certified AI notables, we thought we should give it a little plug. For information, call Craig LaGrow at (415) 957-9353.

RELEASE 1.5

Ingram *Distribution*, not Ingram *Software*, now owns more than 50 percent of Micro D. Ingram Software operates independently of its parent Ingram Distribution, and, of course, of Micro D. This is a sensitive distinction, and we regret the error in our January 28 issue.

PHONE NUMBERS

Frank King, IBM/Austin, (512) 838-0711
Doug Lenat, MCC, (512) 343-0860
Larry Ellison, Oracle, (415) 598-8000
Lee Hecht, Teknowledge, (415) 327-6600
Stan Cornyn, The Record Group, (818) 953-3211

RELEASE 1.0 CALENDAR

March 19	**Consumer Electronics and Retailing Conference** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Terry Earle, (212) 344-8450.
March 20	**Avnet company presentation** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Terry Earle, (212) 344-8450.
March 23-26	**AEA Spring Conference** - Miami. For public companies with less than $75 million in annual revenues. Contact: Flo Lewis at AEA, (415) 857-9300.
March 24	**Warner Computer Systems company presentation** - New York City. Sponsored by the New York Society of Security Analysts. Contact: Terry Earle, (212) 344-8450.
March 24-26	**Office Automation Conference** - Houston. Keynote: Rod Canion. Sponsored by AFIPS. Contact: (800) OAC-1986.
March 24-26	**Interface '86** - Atlanta. Keynote: James Olson, AT&T. Contact: Linda Yogel, Interface Group, (800) 325-3330.
March 25-26	**Connections '86** - San Francisco. Sponsored by Future Computing. Contact: Alice Brown at (214) 437-2400.
March 26	**Boston Computer Society General Meeting** - Boston. Mac creator Jef Raskin talks about Swyftware. Contact BCS at (617) 367-8080.
March 26-27	**Lotus Developers' Conference** - Cambridge, MA. Semi-technical material for third-party developers. Contact: Julie Bingham at Lotus, (617) 577-8500.
March 31-April 1	**Software Tools Conference** - New York. Contact: Warren Briggs at Software Tools, (617) 723-2349.
April 1-3	**Comdex Winter, plus the Software Business Conference** - Los Angeles. Contact: Linda Yogel at the Interface Group, (800) 325-3330.
April 1-4	**First International Conference on Expert Database Systems** - Charleston, SC. Sponsored by IEEE et al. Call Cathie Hughes, (803) 777-5766, or L. Kerschberg, (803) 777-7159.
April 3-6	**West Coast Computer Faire** - San Francisco. Call David Small at Computer Faire, (617) 965-8350.
April 6-7	**Softeach** - Boston. Sponsored by Softsel. Call Kim Kays at (800) 325-9189 or (314) 225-1724.
April 6-9	**ADAPSO Spring Management Conference** - Houston. Call Tom Farewell at ADAPSO, (703) 522-5055.

April 8-10	**Federal Office Systems Expo** - Washington, D.C. Contact: Jackie Voigt at National Trade Productions, (703) 683-8500 or (800) 638-8510.
April 9	**SEMI West Coast Dinner** - San Jose, CA. "Restructuring the U.S. semiconductor industry: Far East marketing." Call Diane Niederer at Semiconductor Equipment and Materials Institute, Inc., (415) 964-5111.
April 9-11	**Corporate Electronic Publishing Systems Show and Conference** - Los Angeles. Call Toni Wiseman or Susan LeDonne at Cahners Exposition Group, (203) 964-0000.
April 13-15	**CD-ROM Based Information Distribution** - Reston, VA. Sponsored by the Institute for Graphic Communication. Contact: Dick Murray at (617) 267-9425.
April 16	**Intel presentation for analysts** - Phoenix. Call Deanna Pavay at (408) 987-5050.
April 16-19	**SPA Spring Symposium** - San Francisco. Contact: Ken Wasch, Software Publishers Association, (202) 452-1600.
April 19-20	**Softeach** - Chicago. Sponsored by Softsel. Call Kim Kays at (800) 325-9189 or (314) 225-1724.
April 21-24	**Sixth International Conference on Decision Support Systems** - Washington, DC. Sponsored by the Institute of Management Science. Call Julie Eldridge, (401) 274-2525.
April 22-23	**Telecommunications Conference** - Boston. Investment seminar sponsored by Alex. Brown & Sons. Call Barbara Murphy at (301) 727-1700.
April 28-30	**Avignon 86** - Avignon, France. Sixth international workshop on expert systems and their applications. Contact: Jean-Claude Rault, Agence de l'Informatique, Tour Fiat - Cedex 16, 92084 Paris-La Defense, France; telephone: (011 331) 47 96 43 14.
April 28-May 1	**Comdex Spring** - Atlanta. Contact: Linda Yogel at the Interface Group, (800) 325-3330.
April 29-May 1	**AI '86** - Long Beach, CA. Contact: Norma Ashman at Tower Conference Management, (312) 668-8100.
May 5-6	**AI Conference** - Stamford, CT. Organized by K. C. Branscomb, (415) 851-4735, in conjunction with the Gartner Group. For registration information call Ashley Pearce at the Gartner Group, (203) 967-6757.
May 6-8	**National Online Meeting** - New York City. Contact: Joanne Loreti at Learned Information, (609) 654-6266.

May 6-8	**National Online Meeting** - New York City. Contact: Joanne Loreti at Learned Information, (609) 654-6266.
May 6-8	**Videotex '86** - Dallas. Contact: Regina Kreischer at Online Conferences, (212) 279-8890.
May 12-13	**Personal Computing Conference** - Nashville. Contact: Ashley Pearce at the Gartner Group, (203) 967-6757.
May 12-14	**OpticalStorage Forum** - Denver, CO. Cosponsored by DISK/TREND and Freeman Associates. Contact: Darlene Plamondon at Catlidge & Associates, (408) 554-6644.
May 14-16	**Info Show** - Los Angeles. Contact: R. Waitzman at Cahners Exposition Group, (203) 964-0000.
May 18-21	**Hammer Forum** - San Francisco. "Shaping the Second Revolution in End-User Computing." Call Pamela Davis at (617) 354-5555.
May 20-21	**PC Products 86: Tech Trends** - Dallas. Sponsored by Future Computing. Contact: Charlene Stevenson at (214) 437-2400.
May 21	**National Data Corporation presentation for analysts** - Atlanta. Call L. C. Whitney's office at the company, (404) 329-8506.
May 28-29	**Engineering and Manufacturing Systems** - New York. Investment seminar sponsored by Alex. Brown & Sons. Call Barbara Murphy at (301) 727-1700.
June 1-4	**Summer Consumer Electronics Show** - Chicago. Contact: Anonymous at Electronic Industries Association/Consumer Electronics Group, (202) 457-8700.
June 3-5	**Optical Storage for Large Systems** - New York City. Sponsored by Rothchild Consultants. Call Judith Hanson at (415) 626-1133.
June 8-11	**World Computing Services Industry Congress** - Toronto. Call Phyllis Cockerham at ADAPSO, (703) 522-5055. (The number given in our previous issue was incorrect.)
June 9-11	**Third Annual Software Industry Conference** - Monterey, CA. Sponsored by the Gartner Group. Contact: Ashley Pearce at (203) 967-6757.
June 10-12	**Comdex International** - Nice, France. Contact: Pete Young at Interface Group, (800) 325-3330, or Charles Vervoord, Amsterdam, (31) 20-6219 41.
June 16-19	**National Computer Conference** - Las Vegas. Sponsored by AFIPS. Contact: (800) NCC-1986.

July 9-11	**PC Expo** - New York City. Contact: Steve Gross at PC Expo, (800) 922-0324 or (201) 569-8542.
July 21-25	**European Conference on Artificial Intelligence '86** - Brighton, UK. Contact: Luc Steels, Vrije Universiteit Brussel, AI Lab, Pleinlaan 2, 1050 Brussels, Belgium; telephone: (011 320) 2-641 2971.
July 22-24	**Future of Optical Memory Technology** - San Francisco. Sponsored by Rothchild Consultants. Call Judith Hanson at (415) 626-1133.
August 6	**IIA Conference** - Cambridge, MA. "Beyond databases: Knowledge delivery systems." Call Linda Cunningham at the Information Industry Association, (202) 639-8260.
August 11-15	**AAAI-86** - Philadelphia. Call Claudia Mazzetti, American Association for Artificial Intelligence, (415) 328-3123.
August 19-22	**International Conference on Parallel Processing** - St. Charles, IL. Sponsored by IEEE and Penn State. Contact: Earl Swartzlander, TRW, (213) 535-4177.
September 2-5	**Comdex/Australia** - Sydney. Yet another Comdex. Call Linda Yogel at the Interface Group, (800) 325-3330.
September 7-10	**ABCD Breakway '86** - Lake Buena Vista (Orlando), FL. The third annual convention of the Association of Better Computer Dealers. Contact: Terri Rojas at (312) 693-ABCD.
September 8-10	**NCC Telecommunications '86** - Philadelphia. Sponsored by AFIPS. Contact: (800) NCC-1986.
October 20-22	**UNIX Expo** - New York City. Contact: Don Berey or Bob Birkfield at National Exposition Co., (212) 391-9111.
October 27-29	**Interface International** - Paris. Contact: Linda Yogel, Interface Group, (617) 449-6600.
November 2-5	**ADAPSO 65th Management Conference** - Phoenix. Call Phyllis Cockerham at ADAPSO, (703) 522-5055.
November 10-14	**Comdex Fall** - Las Vegas. Contact: Linda Yogel, Interface Group, (617) 449-6600.

1987

February 22-25	**Tenth Annual Personal Computer Forum** - Phoenix. For information, call Sylvia Franklin at EDventure Holdings/ Release 1.0, (212) 503-5517.

Please let us know of any other significant events we should include.

-- Laurin Henchey

SUBSCRIPTION FORM

Please enter my subscription to **Release 1.0** at the rate of $395 per year in the U.S. and Canada. Overseas subscriptions are $475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request.

Name_____

Title_____

Company_____

Address_____

City_____ State_____ Zip_____

Telephone_____

Please fill in the information above and send with your payment to:

Sylvia Franklin
Release 1.0
Ziff-Davis Publishing Company
One Park Avenue
3rd Floor
New York, NY 10016

If you have any questions, please call us at (212) 503-5517.